The Report Committee for James Wesley Freeman
Certifies that this is the approved version of the following report:

Using EM Algorithm to Identify Defective Parts Per Million on Shifting Production Process

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Dragan Djurdjanovic

Matt Hersh

Using EM Algorithm to Identify Defective Parts Per Million on Shifting Production Process

by

James Wesley Freeman, B.S.As.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Statistics

The University of Texas at Austin

December 2012

Acknowledgements

Primarily I am grateful to the Lord for giving me the strength and stamina to work, have a family, and pursue a higher education. Without Devine intervention I do not believe I could have accomplished it.

Secondly, I would like to thank my wife and five children for patiently waiting and supporting me through these last few years. There have been many missed camping trips, family events, etc. endured while I worked at home completing homework, writing papers, and preparing for exams. My wife, Melissa, deserves the largest commendation and as much credit as I for reaching this point. She continually encouraged me, acted as a single parent and persevered with homeschooling all our children.

Thirdly, I would like to thank my managers at Applied Materials, Hereaus Quartz, and 3M for allowing me to take off from work during inconvenient times to attend class and professors' office hours. Without their cooperation, I would not have had this opportunity.

Last but certainly not least, I would like to thank the two professors I have worked with during the course of the program and for offering their assistance in the writing of this paper – Dr. Dragan Djurdjanovic and Dr. Matthew Hersh.

Abstract

Using EM Algorithm to Identify Defective Parts Per Million on Shifting Production Process

James Wesley Freeman, M.S. Stat.

The University of Texas at Austin, 2012

Supervisor: Dragan Djurdjanovic

The objective of this project is to determine whether utilizing an EM Algorithm to fit a Gaussian mixed model distribution model provides needed accuracy in identifying the number of defective parts per million when the overall population is made up of multiple independent runs or lots. The other option is approximating using standard software tools and common known techniques available to a process, industrial or quality engineer. These tools and techniques provide methods utilizing familiar distributions and statistical process control methods widely understood. This paper compares these common methods with an EM Algorithm programmed in R using a dataset of actual measurements for length of manufactured product.

Table of Contents

List of Tables

List of Figures

CHAPTER 1: INTRODUCTION

PROJECT MOTIVATION

Original Equipment Manufacturers (OEMs) will often design a product and then have a contract manufacturer produce it. By doing so, contract manufacturers can reduce the OEM's production cost and provide flexibility in the production process (Cheng, Pg 889). When a contract manufacturer provides this product, the OEM having designed the product is expecting the supplier to meet all requirements detailed by the product specifications. To verify the supplier is meeting the specifications, the OEM may require the supplier to provide test and inspection data. The test and inspection data is usually based upon results from the final product. This data may be monitored by the OEM for accessing how well the product and process is being controlled relative to the specifications.

In some cases, the first tier supplier may not be mature in process control and thus not monitoring the input variables to the process. This lack of monitoring of input variables prevents an ability of the contract manufacturer to control the output. Therefore, both the contract manufacturer and OEM only know the results from the process without having a cause and effect understanding.

The lack of input variable information puts the OEM's engineer, who is monitoring the supplier's process data, into a limited visibility situation. Neither the OEM or contract manufacturer is willing to make a financial investment to understand or reduce variation in the process.[1] The engineer though is still expected to predict process performance. These individuals monitoring the product are interested in predicting how

[1] The author knows of some companies in the semi-conductor industry that do expect their first tier suppliers to understand their process variation and to spend effort to reduce its amount.

many units may not meet a specific characteristic. If a process runs continuously or stopped and started with no extraneous variables entering the restart, the output measurements can be modeled using a single probability distribution such as Gaussian, Weibull, etc.

The difficulty arises when production is run for a part, stopped, set up differently to run a different part and after several cycles, the original part is run again. These type runs can be seen in plastic molding facilities, metal machining operations, and other industries where limited numbers of components are ran to meet order quantities. The population parameters may shift because of a variable not present on one run is present on the next run. This variation may be due to a latent variable associated with operator changes, environmental changes, raw material changes, etc.

These type runs make it difficult to assess the proportion of parts which can be expected to fall outside specifications. If each run is evaluated individually, the measurements of the part characteristic will usually have a reasonable density distribution. However, when the individual values from run to run are compiled into a single dataset, the dataset has a non-parametric distribution.

This lack of a recognizable distribution does not relieve the engineer of the responsibility to provide the OEM's management team the predicted loss of yield due to product not within specifications. Several mechanisms to model the predicted loss can be used. One method is to pick a distribution that most closely represents the compiled data but may not be an accurate fit. A second methodology is to count the number of past failures and create a percentage outside the range. A third method is to use a Mixed Distribution Model.

The objective of this project is to use all three methodologies to compare their accuracy in predicting the out of specification likelihoods.

PAPER OVERVIEW

The paper leads off with an explanation of how manufacturing processes are monitored based upon the author's experience. The paper then discusses the Expectation Maximization (EM) algorithm theory. After the theory is discussed, different techniques are applied to the data for manufactured product. Once the models and expectations are created for most of the lots, the remaining lots are used to validate the models. A summary is then made detailing the findings and whether the need is required for more advanced analysis and prediction. Following the summary are some future additional areas of study the author can pursue to determine if the more complex methods begin to show more benefit than the simpler methods.

Chapter 2: Common Manufacturing Process Monitoring

Methods utilized for process control and analysis in manufacturing facilities are dependent upon common understood practices such as statistical process control (SPC). These practices and methodologies are readily available in statistical process control reference books. These common understood practices are assumed to be Gaussian distributions. The results are approximately correct even if the distribution is not normal. (Montgomery, Pg 203).

The primary tools used are the calculation of Cpk, defective parts per million (DPPM), and the general percent defective. The equations are listed in Equations 1-3 respectively.

$$C_{pk} = \frac{\text{Min}[(\text{Upper Spec.Limit} - \bar{X}) \text{or } (\bar{X} - \text{Lower Spec.Limit})]}{3\sigma} \quad \text{(Eq. 1)}$$

$$DPPM = \frac{Portion\ Outside\ Specification\ Range}{Total\ Number\ of\ Units\ Produced} * 1{,}000{,}000 \quad \text{(Eq. 2)}$$

$$Percent\ Defective = \frac{Portion\ Outside\ Specification\ Range}{Total\ Number\ of\ Units\ Produced} * 100 \quad \text{(Eq. 3)}$$

Chapter 3: EM Algorithm

One common technique used to identify if multiple populations exist within the manufacturing data set is the Expectation Maximization (EM) algorithm. The EM algorithm is an iterative procedure for handling missing data. The method alternates between an imputation step and an analysis step to solve for the parameter estimates (Pearson, Pg 723). In general, it finds the maximum-likelihood estimate of parameters for an underlying distribution from a given data set when the data is incomplete or has missing values (Bilmes, Pg 1).

At a rudimentary level, a single set of data with no missing data has a resulting joint density function shown in Equation 4.

$$p(X|\mu,\sigma) = \prod_{i=1}^{N} N(x_i|\mu,\sigma) = \mathcal{L}(\mu,\sigma|X) \qquad \text{(Eq.4)}$$

This joint density function is also called the likelihood function or in other words the likelihood of the parameters given the data. If data is missing however, it is not possible to solve and find analytical expressions for the density function parameters. It would be considered an incomplete-data likelihood function. Therefore, more complex techniques are required (Bilmes, Pg 1).

Missing values can occur in a joint distribution *Z=(X,Y)* where values of *X* are observed and some of the Y observations are missing or where *N-j* random numbers of set $\{x_i\}$ are observed and the last j random numbers are missing.

In the case of Z=(X,Y), the complete data set exists as Z=(X,Y) and the joint Gaussian density function becomes Equation 5 using conditional probability.

$$p(z|\mu,\sigma) = p(x,y|\mu,\sigma) = p(y|x,\mu,\sigma) * p(x|\mu,\sigma) = \mathcal{L}(\mu,\sigma|\mathcal{Z}) \qquad \text{(Eq. 5)}$$

$p(x, y|\mu, \sigma)$ is the complete data likelihood function, $p(y|x, \mu, \sigma)$ is the conditional distribution of Y and $p(x|\mu, \sigma)$ is the marginal distribution of X. This joint distribution of the present and missing data creates a new likelihood function called the complete-data likelihood (Bilmes, Pg 2).

The EM algorithm first finds the expected value, the E step, of the complete-data log likelihood "$\log p(x, y|\Theta)$" with respect to the unknown data Y given the observed data X and the current parameter estimates. This function appears as Equation 6 (Bilmes, Pg 2).

$$E\left[\log\ p(x, y|\mu, \sigma)|x, \mu^{(i-1)}\sigma^{(i-1)}\right] =$$

$$\int_{y\epsilon\Upsilon} \log p(x, y|\mu, \sigma) * f(y|x, \mu^{(\mathrm{i}-1)}\sigma^{(\mathrm{i}-1)})\, dy \qquad \text{(Eq. 6)}$$

Υ is the value y can take on and $p(y|x, \mu^{(\mathrm{i}-1)}\sigma^{(\mathrm{i}-1)})$ is the marginal distribution of the unobserved data and is dependent on both the observed data $\{x_i\}$ and the current parameters μ and σ. The second step (the M-step) of the EM algorithm then computes the maximum likelihood estimates of the parameters in question[2].

Once the new parameter estimates are computed from the old expected parameter estimates, the difference between the log likelihood values from iteration j+n to j+n+1 is computed and convergence is checked. Once the log likelihood has converged, the process is terminated.

The manufacturing process data being evaluated for population shifts does not have data missing from a single data set but is considered a mixture of probability

[2] See Pearson page 724 – 726 for an example of EM Algorithm being mathematically calculated for a simple bivariate example.

densities. The intent is for the EM algorithm to evaluate the data to identify parameters for each of the distributions. The missing data is considered the data for the latent variable causing the different probability density functions or in other words the different populations. This type of problem is possibly one of the most widely used applications of the EM algorithm in the computational pattern recognition community (Bilmes, Pg 3).

Equation 7 is the function for the mixed probability density model.

$p(x) = \sum_{i=1}^{M} \alpha_i N(x|\mu_i \sigma_i)$ where $\sum_{i=1}^{M} \alpha_i = 1.$ (Eq. 7)

The α term is the probability of an individual probability density occurring. As in the previous case, an incomplete-data log-likelihood expression is created for a mixed Gaussian model. The expectation expression is shown in Equation 8. Notice the difference is the α term has been added for the different populations.

$$E\left[\log p(X, Y|\alpha, \mu, \sigma)|X, \alpha^{(i-1)}, \mu^{(i-1)}, \sigma^{(i-1)}\right] =$$

$$\int_{y \epsilon Y} \log p\,(X, y|\alpha, \mu, \sigma) * f(y|X, \alpha^{(i-1)}, \mu^{(i-1)}, \sigma^{(i-1)}\, dy \quad \text{(Eq. 8)}$$

This expression is maximized for α, μ, and σ. The mathematics required to allow for the maximization activity requires using Baye's rule and a Lagrange multiplier. (Bilmes, Pg 3-5). Bilmes, pp 3-5 and Bishop, pp 430 – 439 makes this derivation. The final parameter estimates are summarized in Equations 9-11 (Bishop, pg 439).

$$\alpha_l^i = \frac{N_l}{N} \quad \text{(Eq. 9)}$$

$$\mu_l^i = \frac{1}{N_l} \sum_{i=1}^{N} \left(\frac{\alpha_l N(x_i|\mu_i, \sigma_i)}{\sum_{j=1}^{K} \alpha_j N(x_i|\mu_j \sigma_j)} \right) * x_i \quad \text{(Eq. 10)}$$

$$\sigma_l^i = \frac{1}{N_l} \sum_{i=1}^{N} \left[\left(\frac{\alpha_l N(x_i|\mu_i, \sigma_i)}{\sum_{j=1}^{K} \alpha_j N(x_i|\mu_j \sigma_j)} \right) \left(x_i - \mu_l^i\right) \left(x_i - \mu_l^{(i-1)}\right)^T \right] \quad \text{(Eq. 11)}$$

where $N_l = \sum_{i=1}^{N} \frac{\alpha_l N(x_i|\mu_i,\sigma_i)}{\sum_{j=1}^{K} \alpha_j N(x_i|\mu_j\sigma_j)}$

The process for solving for solving for the parameters is now the same as in the simpler case of having a single Gaussian distribution with missing data. These equations perform the expectation and maximization step simultaneously. The algorithm proceeds by using the newly derived parameters as the guess for the next iteration The log likelihood is then re-evaluated. (Bilmes, Pg. 7).

Figure 1: Log Likelihood Values for Manufacturing Data

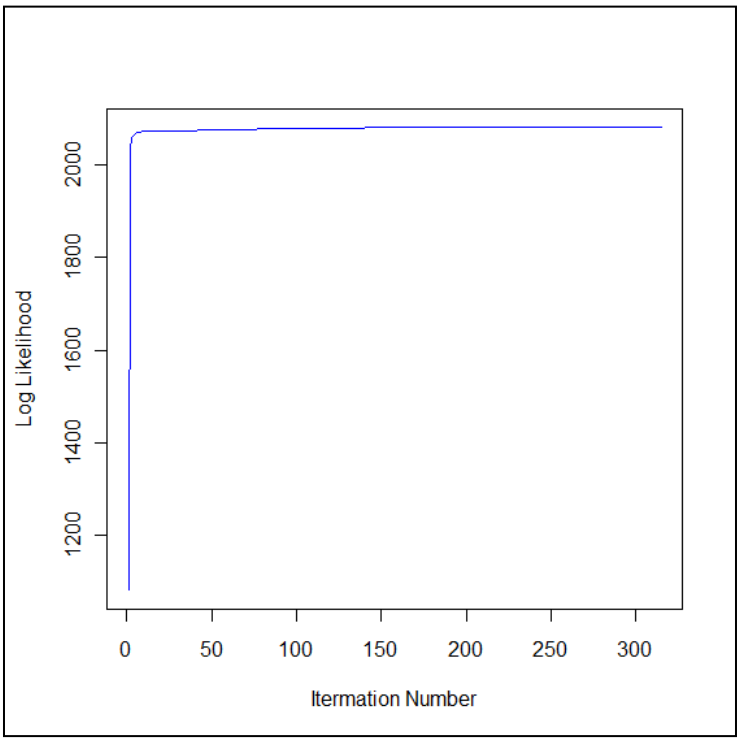
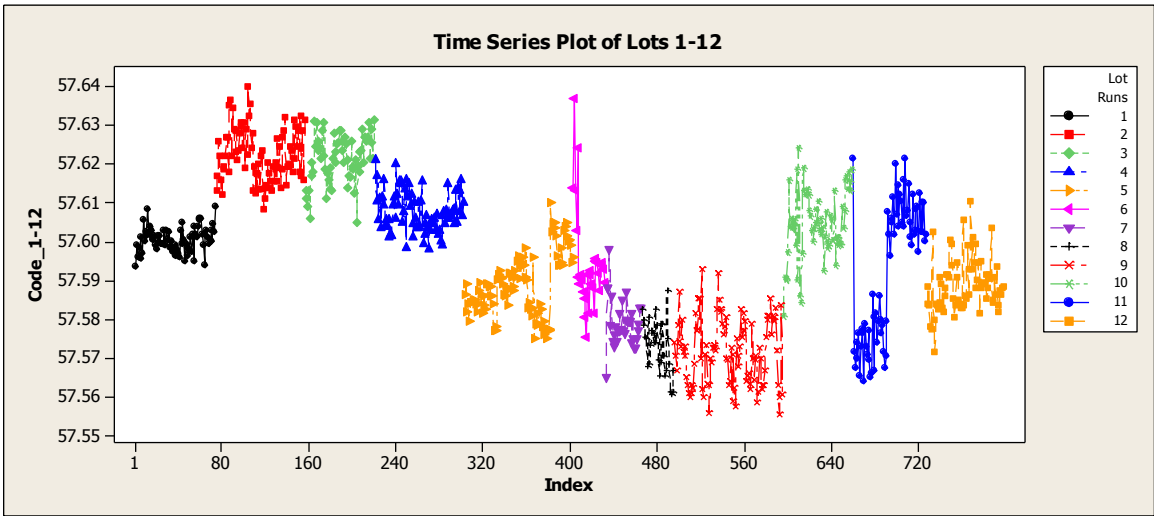


Figure 1 above shows the log likelihood values from the EM algorithm for lots one to twelve of the manufacturing data to be discussed in Chapter 4. Figure 2 shows the lots the EM Algorithm function will use to determine α_l, μ_l, σ_l. The quantity per run varies from 30 to 100 points. Not only is there the appearance of population shifts

between lots but also within lots. The cause or causes of the shifts between populations is not visible to the OEM.

Figure 2: Length Measurements for Lots 1-12



Chapter 4: Final Model Results and Findings

Analysis of Initial 12 Runs – Gaussian

The data for twelve lots was analyzed and the results in Figure 3 were produced. The results for overall performance indicate 2.4% of the values will be above the upper specification limit and 10.0% will be below the upper specification limit. Confirmation by a Q-Q Plot in Figure 4 shows the data is not a Gaussian distribution. The population has heavier tails. Even though the data does not adhere to a Gaussian distribution, an engineer will report the contract manufacturer may have approximately 12.4% or 124,000 DPPM units out of specification. This percentile information is identified from the Expected Overall Performance indicated in parts per million from Figure 3. If a run consists of 500 units manufactured, then 62 units are expected to be outside the specification.

Figure 3: DPPM Results for Lots 1-12

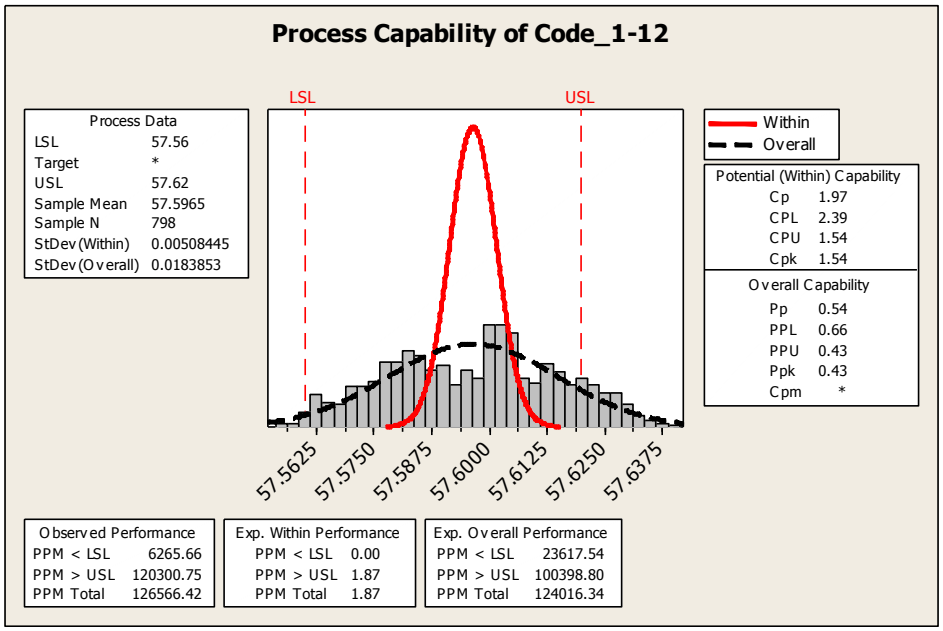
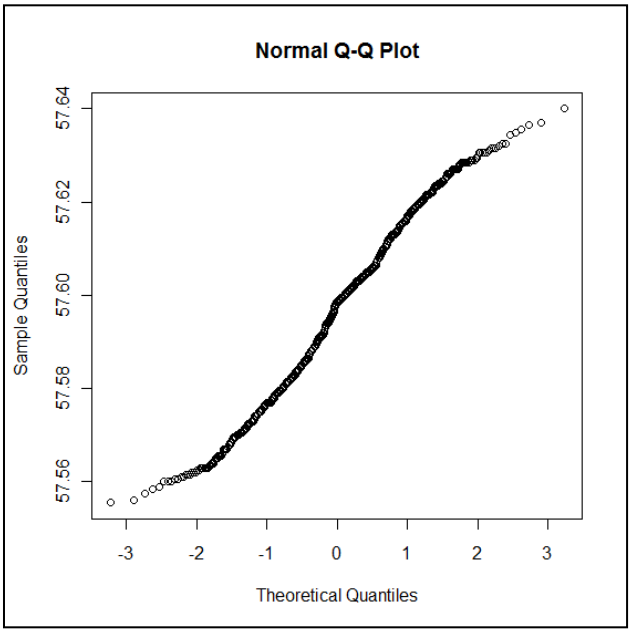


Figure 4: Q-Q plot of Lots 1-12



ANALYSIS OF INITIAL 12 RUNS – BINOMIAL APPROXIMATION

Given the distribution does not lend itself to a normal distribution; an engineer may look at it based on a binomial density function of a failure from the data. There are 798 units, 97 measure outside the specification window. These values yield a failure probability of 0.122. This probability provides no insight though. A 95% confidence interval for the mean states the process could produce as many as 14.7% failures or as few as 10.0% failures on average. These values do not provide insight into which side of the specification window the units may fall on. On some component characteristics, this may be important as to how the material is identified for rework or scrap.

Figure 5: Binomial Failure Rate for a 500 Unit Run

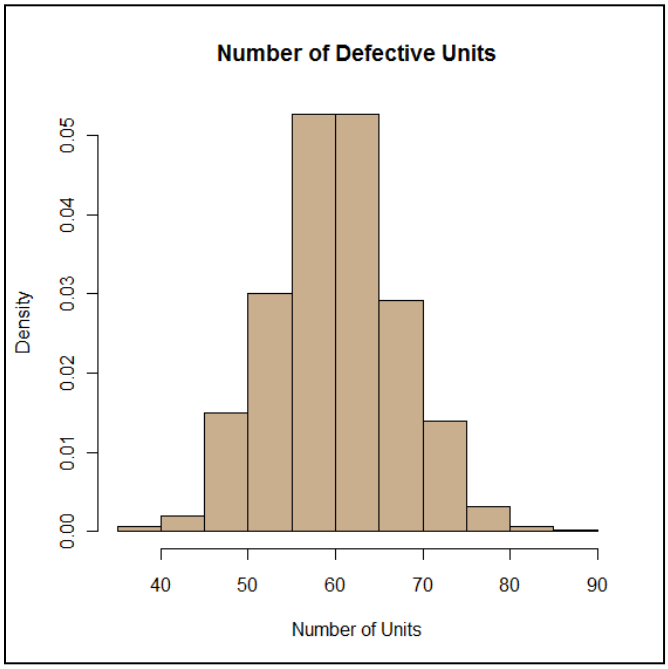


Figure 5 is the mass density function for a binomial with a 12.1% success rate for a 500 unit run. The number of failing units fall within a 95% interval of 49 to 73.

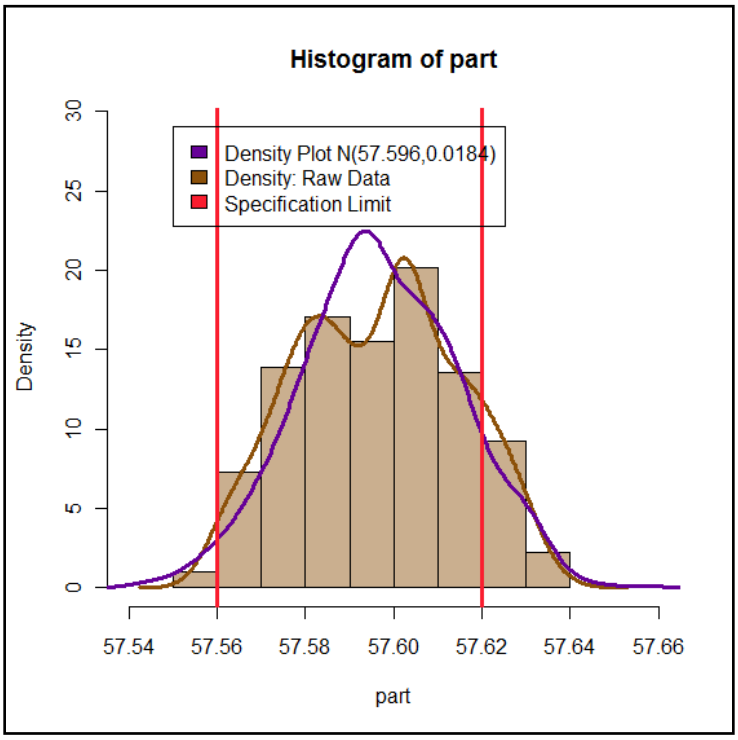
Analysis of Initial 12 Runs – EM Algorithm with Gaussian Mixed Distributions

The overall histogram of the first twelve production runs is shown in Figure 6. Along with the histogram, the specifications limits are visible along with two density plots. One density plot is taking the overall average and standard deviation of the data and fitting a Gaussian distribution to the data. Using a $\mathcal{N}(57.596, 0.0184)$ density function, ten percent of the units are above 57.62 and 2% are below 57.56[3] which aligns

[3] The code for identifying the percentages above and below the specification is in Appendix B -Figure 6 Histogram having variable "upercent" and "lpercent".

with the prediction from Figure 3.The second density plot is of the individual values. It shows two modes along with a possible third on its right side.

Figure 6: Histogram of First Twelve Production Runs



The methodology for the EM Algorithm in R programming code is devised as follows. The *normalmixEM* function within the "mixtools" package (See Appendix A) will be used to identify the number of populations within the overall dataset of lots one to twelve. A second routine using a bootstrap methodology also within the "mixtools" package will corroborate or recommend an alternate amount of population components.

The EM Algorithm method identifies two populations. One population has a 0.639 probability of occurring (green density function in Figure 7) and the other has a 0.36 probability of occurring (red density function in Figure 7). This information allows an engineer to identify the possible population the process will generate and then assign a

Gaussian distribution for the respective population. Figure 7 below shows the two populations within the overall dataset. Table 1 contains the derived statistics for each of the Gaussian probability density functions.

Figure 7: Gaussian Mixed Distribution Plot with Production Data

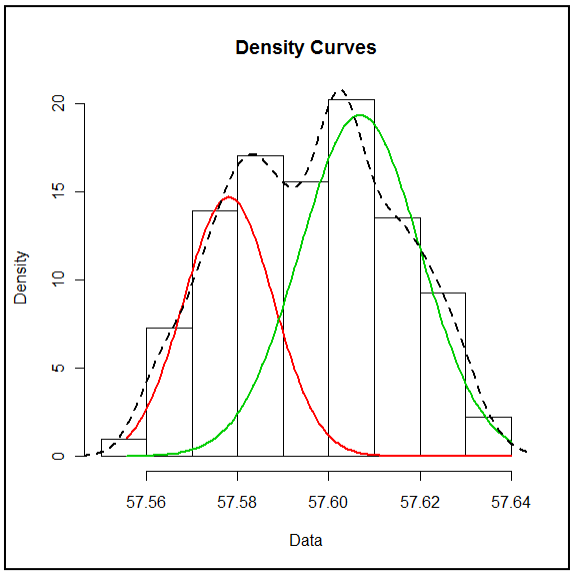


Table 1: Derived Statistics for 2 Components Mixed Gaussian Distribution

K=2	Probability	Mean	Std Dev
Population 1	0.361	57.578	0.0098
Population 2	0.639	57.607	0.0132

A method for verifying if a two component model is adequate is to perform hypothesis testing on the number of Gaussian components that is populations. A second function *boot.comp*, also from the "mixtools" package, performs the hypothesis test. This function produces X "bootstrap realizations of the likelihood ratio statistic for testing the null hypothesis of a k-component fit versus the alternative hypothesis of a (k+1)-component fit to various mixture models" as stated in R Documentation[7]. The results

of this function indicate two populations are the correct number of populations to utilize. Table 2 provides the results from the function. The p-value for going from a two component Gaussian model to a three component Gaussian model is not significant.

Table 2: Hypothesis Test Results for Gaussian Components

	1 component vs 2 components	2 components vs 3 components
p-value	0	0.14
log likelihood test statistic	45.2	33.7

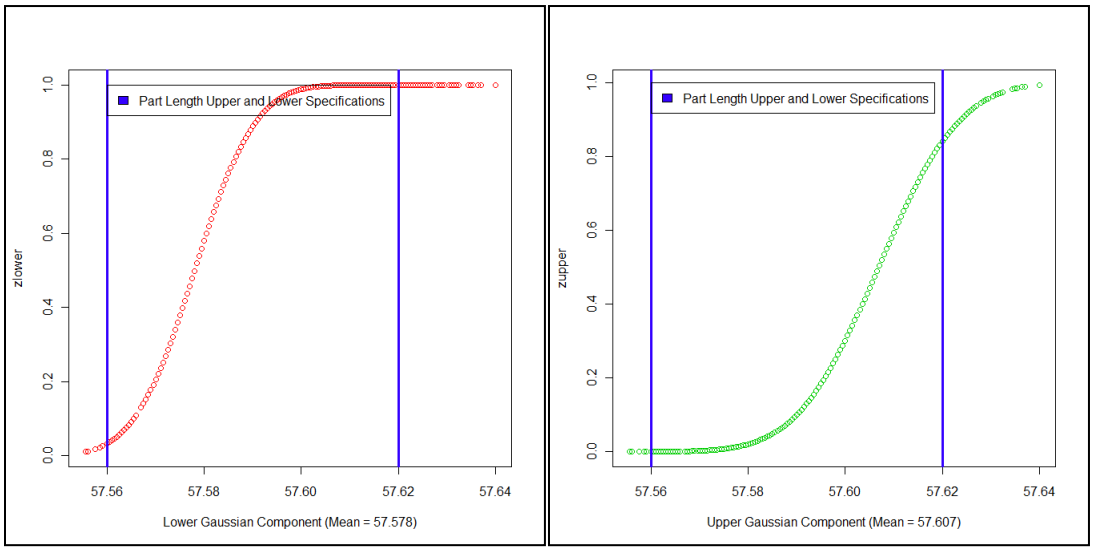
As a third informal check, the *normalmix EM* function was run 100 times. The additional runs were used to verify no more than two components were identified and whether the statistics for the Gaussian components converged to consistent values. Of the 100 runs, only one did not converge. These statistical estimates for parameters were not used. All other ninety-nine runs had similar values to the forth place digit for the mean and fifth place digit for the standard deviation. Appendix D has the tabulated results for the 100 runs.

With the number of populations identified and their respective mean and standard deviations known, the mixed model was identified. The mean was specified to the third place digit and the standard deviation to the fifth place digit, Table 3 indicates the percentage above and below the part length specifications depending upon which population the part length falls within. The proportion outside the specification limits was calculated using the actual measurement data and the cumulative function for each of the Gaussian distributions, Figure 8.

Table 3: Predicted Percentages Outside Specifications

	Mean	Standard Deviation	Proportion Below 57.56	Proportion Above 56.62
Component 1	57.578	0.0098	0.0328	0.0000
Component 2	57.607	0.0132	0.0002	0.1600

Figure 8: Cumulative Functions - Gaussian Component

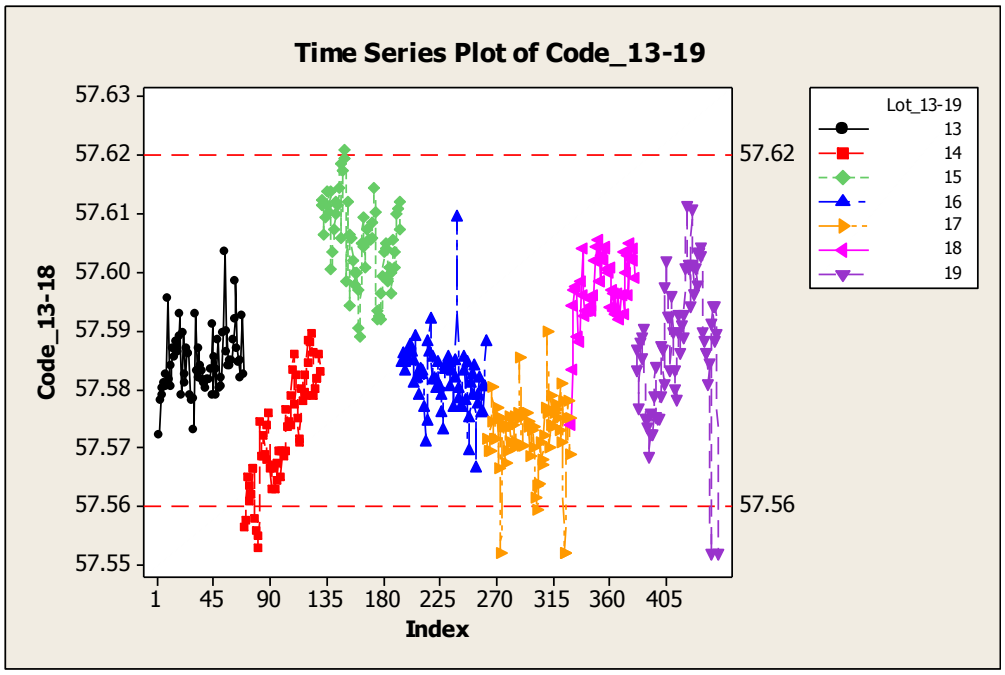


To rephrase the analysis, the manufacturing process has an approximate 3:2 propensity to manufacture parts with a mean of 57.607 versus 57.578 and standard deviation of 0.0132 versus 0.0098. When the $\mathcal{N}(57.607, 0.0132)$ population occurs, an approximate proportion of 0.16 will be out of specification. If the alternate population $\mathcal{N}(57.578, 0.0098)$ occurs, an approximate proportion of 0.03 will be out of specification. This type of information provides more information around the process but takes more time to generate.

FINAL MODEL FINDINGS – TESTING

From the previous section, an engineer was able to generate several proportions from models predicting the propensity of the process to generate out of specification units. The prediction models will now be compared against the lots thirteen through nineteen. Figure 9 shows the length measurements along with the upper and lower specifications.

Figure 9: Scatter Plot for Lots 13-19



The first two model predictions use the Gaussian and Binomial models. Applying the results of training the models in the previous section to the later lots 13-19 yields the following results in Table 4. The Gaussian approximation largely over predicts the quantity going outside the upper specification limit. To both companies, the OEM and contract manufacturer, the error in the model is not advantageous. If the run rate was to overproduce because of this expected fallout, too many components would have been produced and thus shelved in inventory. The same type of scenario is present for the

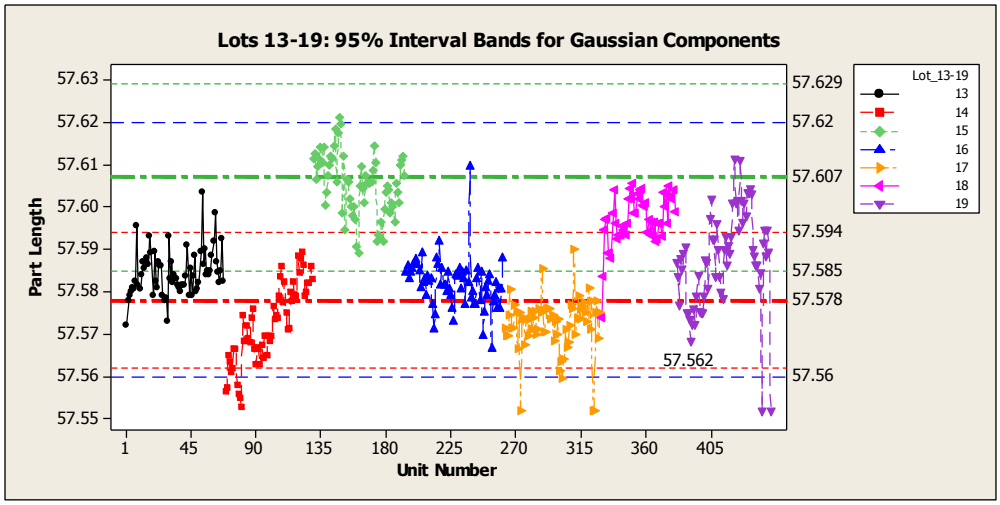
Binomial approximation. The 95% interval would have predicted 43 units as a minimum would have failed. In this case, only 13 units failed. Again over production would have resulted.

Table 4: Gaussian and Binomial Models

Method	Predicted Below Specification	Predicted Above Specification	Lot 13-19 Actual Below Specification	Lots 13-19 Actual Above Specification
Gaussian Approximation	2.40%	10%	2.69%	0.22%
Binomial Appoximation*	12.1% failure, Expect 43 to 65		2.9% failure, 13 units failed	
*Note: 446 units run				

Based upon the results in Table 4, the additional time to generate the EM Algorithm is advantageous. Even though the propensity of the model is to have approximately 6 out of 10 runs around the upper distribution, the engineer can gage which distribution is about to be generated by knowing the first few data points of the run. For lots thirteen through nineteen, if the first few data points are compared relative to the 95% interval for each of the Gaussian components, the lower Gaussian component would be chosen. The engineer would have estimated 3.28% would fail. The actual failure was 2.69% failure.

Figure 10: Lots 13 - 19, Gaussian Components with 95% Intervals



Chapter 5: Summary & Future Studies

SUMMARY

With the aim of the investigation to determine if a Gaussian approximation of non-Gaussian data provides a better model, the answer is yes. However, the answer yes comes with a caveat. The caveat is in this case the volume is small. From a manufacturing perspective, over producing by one or two parts due to a difference in 2% versus 3% is not a dramatic difference. However, if on a volume scale such as the manufacturing of semiconductor chips, this small difference can be profound in how many units are produced.

Another factor to be considered is the cost of a unit. If the cost of over producing and maintaining the cost of inventory costs more than taking the time to have more accurate models, consideration should be given to investing time into training engineers for developing more accurate models. However if the production is low volume, low cost units, a fifteen second approximation is all that is required and no further analysis is needed. Each engineer and each company has to evaluate this requirement and come to their own conclusion.

FUTURE STUDIES

Future studies can be performed on additional data sets in which the Gaussian components begin to diverge from each other. By having the Gaussian distributions begin to diverge, the accuracy of the single Gaussian approximation will begin to

breakdown. At what point the does the approximation no longer provides a descent approximation, relatively speaking, for low volume manufacturing.

In addition, each production run could be evaluated using EM Algorithm technique. In this case though, a higher run rate would be needed to justify the amount of time and logistical work for machine and operator feedback.

Appendices

Appendix A – R Code: Mixtools

The R Code "mixtools" The mixtools package is available from the Comprehensive R Archive Network at http://CRAN.R-project.org/package=mixtools. The package was published and discussed in The Journal of Statistical Software, October 2009, Volume 32, Issue 6. The article was written by Tatiana Benaglia - Pennsylvania State University, Didier Chauveau – Universit*é* d'Orl*é*ans, David R. Hunter - Pennsylvania State University, and Derek S. Young - Pennsylvania State University. The function "normalmixEM" was utilized from this package.

Appendix B – R Code Created for Analysis

Initial Code to read in the data

```
#Load the data
part<-read.csv("C:/Users/a2y3yzz/Documents/Wesley/Mfg Paper/Final Analysis Code and
Data/part_data.csv",header=T)

#Load the mixtools library
library(mixtools)

#Return the matrix obtained by converting all the variables in a data frame to numeric mode
part<-data.matrix(part)
```

Error! Reference source not found.

```
----------Figure in EM Algorithm Section ----------
#verify loglik location in mixmdl
plot(mixmdl, which=1)
w=mixmdl$all.loglik
w1=w[-(201:315)]
plot(w,type='l',col="blue",xlab="Itermation Number", ylab="Log Likelihood")
```

Figure 5 – Binomial Approximation Code

```
set.seed(101)
#generates 1000 opportunities (i.e. numbers)
#500 trails per opportunity
#0.121 success rate per trial
hist(rbinom(1000,500,0.121),col='#8C510A75',xlab=' ', ylab=' ',main=' ',freq=FALSE)
title(main='Number of Defective Units', xlab='Number of Units', ylab='Density')
lower_05_unit_bound<-qbinom(0.05,500,0.121)
lower_05_unit_bound
upper_95_unit_bound<-qbinom(0.95,500,0.121)
upper_95_unit_bound
```

Figure 6 Histogram

```
mean.method1<-mean(part)
sd.method1<-sd(part)
hist(part)
hist(part,freq=FALSE,xlim=c(57.54,57.66),ylim=c(0,29),col='#8C510A75')
points(density(part),col='#8C510A',type='l',lwd=3)
points(density(rnorm(798,mean.method1,sd.method1)),col='#660198',type='l',lwd=3)
upercent<-pnorm(57.62,mean=mean.method1,sd=sd.method1)
lpercent<-pnorm(57.56,mean=mean.method1,sd=sd.method1)
abline(v=57.62,lwd=3,col="#FA1D2F") #lwd is the the width of the line
abline(v=57.56,lwd=3,col="#FA1D2F") #lwd is the the width of the line
```

```
legend(57.55,29,legend=c('Density Plot N(57.596,0.0184)','Density: Raw Data','Specification
Limit'),fill=c('#660198', '#8C510A','#FA1D2F'))
```

R code for binomial confidence interval

```
#Binomial test for 95% confidence interval
#not interested in the actual test but part of the output is the confidence interval around the failure
percentage
prop.test(97,798)
```

R code for EM Algorithm

```
#Mixed Model Algorithm to identify the lambda, mu, and sigma parameters - Single Run
set.seed(101)
mixmdl=normalmixEM(part)
plot(mixmdl,which=2)
lines(density(part),lty=2,lwd=2)
summary(mixmdl)
```

R code for EM Algorithm Iteration

```
#----------Iteration for Gaussian Mixed Models----------
times<-100
L<-matrix(NA,ncol=10,nrow=times)
M<-matrix(NA,ncol=10,nrow=times)
S<-matrix(NA,ncol=10,nrow=times)
for(i in 1:times){
mixmdl.i<-normalmixEM(part)
n.i<-length(mixmdl.i$lambda)
L[i,1:n.i]<-mixmdl.i$lambda
M[i,1:n.i]<-mixmdl.i$mu
S[i,1:n.i]<-mixmdl.i$sigma
row}
```

Table 3 Code

```
#----------------Identifying the percentage using the smaller population distribution----------------
zlower<-pnorm(part,mean=mixmdl$mu[2],sd=mixmdl$sigma[2])
mixmdl$mu[2]
par(mfrow = c(1, 1))
plot(part,zlower,col="#FF0000",xlab='Lower Gaussian Component (Mean = 57.578)')
lower.dist<-cbind(part,zlower)
lower.dist[order(lower.dist[,2]),]
abline(v=57.62,lwd=3,col="#3300FF") #lwd is the the width of the line
abline(v=57.56,lwd=3,col="#3300FF") #lwd is the the width of the line
legend(57.56,1.0,legend=c('Part Length Upper and Lower Specifications'),fill=c('#3300FF'))

#----------------Identifying the percentage using the upper population distribution----------------
zupper<-pnorm(part,mean=mixmdl$mu[1],sd=mixmdl$sigma[1])
mixmdl$mu[1]
plot(part,zupper,col="#00CD00",xlab='Upper Gaussian Component (Mean = 57.607)')
upper.dist<-cbind(part,zupper)
upper.dist[order(lower.dist[,1]),]
abline(v=57.62,lwd=3,col="#3300FF") #lwd is the the width of the line
abline(v=57.56,lwd=3,col="#3300FF") #lwd is the the width of the line
legend(57.56,1.0,legend=c('Part Length Upper and Lower Specifications'),fill=c('#3300FF'))
```

Appendix C – Data Set

Lots 1-12

part_data.csv

Lots 13-19

part_data_13-19.csv

Appendix D – K=2, 100 iterations

	Population Probability		Means		Std Deviation		
Number of Iterations	k=1	k=2	k=1	k=2	k=1	k=2	Comments
294	0.3605907	0.6394093	57.57805	57.60687	0.009801	0.013201	
293	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
174	0.3606446	0.6393554	57.57805	57.60687	0.009801	0.0132	
297	0.3605908	0.6394092	57.57805	57.60687	0.009801	0.013201	
212	0.3606451	0.6393549	57.57805	57.60687	0.009801	0.0132	
273	0.360645	0.639355	57.57805	57.60687	0.009801	0.0132	
205	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
320	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
206	0.360645	0.639355	57.57805	57.60687	0.009801	0.0132	
284	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
204	0.3606453	0.6393547	57.57805	57.60687	0.009801	0.0132	
219	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
326	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
254	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
161	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
182	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
218	0.3606445	0.6393555	57.57805	57.60687	0.009801	0.0132	
250	0.3605907	0.6394093	57.57805	57.60687	0.009801	0.013201	
250	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
184	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
205	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
254	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
225	0.3605903	0.6394097	57.57805	57.60687	0.009801	0.013201	
204	0.3605904	0.6394096	57.57805	57.60687	0.009801	0.013201	
281	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
249	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
273	0.3606455	0.6393545	57.57805	57.60687	0.009801	0.0132	
378	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
234	0.3605906	0.6394094	57.57805	57.60687	0.009801	0.013201	
240	0.360645	0.639355	57.57805	57.60687	0.009801	0.0132	
452	0.3605902	0.6394098	57.57805	57.60687	0.009801	0.013201	
213	0.3606453	0.6393547	57.57805	57.60687	0.009801	0.0132	
204	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
1000	0.5528509	0.4471491	57.59566	57.5975	0.018364	0.018335	Failed to Converge - That is, the unit failed to meet the log-likelihood delta from one iteration to the next after 1000 iterations.
381	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
230	0.3605903	0.6394097	57.57805	57.60687	0.009801	0.013201	
225	0.3605906	0.6394094	57.57805	57.60687	0.009801	0.013201	
246	0.3605911	0.6394089	57.57805	57.60687	0.009801	0.013201	
224	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
205	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
198	0.360591	0.639409	57.57805	57.60687	0.009801	0.013201	
326	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
365	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
245	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
193	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
281	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
250	0.360645	0.639355	57.57805	57.60687	0.009801	0.0132	
211	0.3606445	0.6393555	57.57805	57.60687	0.009801	0.0132	

	Population Probability		Means		Std Deviation		
Number of Iterations	k=1	k=2	k=1	k=2	k=1	k=2	Comments
287	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
233	0.3606446	0.6393554	57.57805	57.60687	0.009801	0.0132	
208	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
273	0.3606451	0.6393549	57.57805	57.60687	0.009801	0.0132	
179	0.3605908	0.6394092	57.57805	57.60687	0.009801	0.013201	
219	0.3605902	0.6394098	57.57805	57.60687	0.009801	0.013201	
183	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
409	0.3606455	0.6393545	57.57805	57.60687	0.009801	0.0132	
219	0.3605903	0.6394097	57.57805	57.60687	0.009801	0.013201	
314	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
384	0.3606451	0.6393549	57.57805	57.60687	0.009801	0.0132	
255	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
189	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
315	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
151	0.3606447	0.6393553	57.57805	57.60687	0.009801	0.0132	
310	0.3606455	0.6393545	57.57805	57.60687	0.009801	0.0132	
230	0.3605903	0.6394097	57.57805	57.60687	0.009801	0.013201	
186	0.3605904	0.6394096	57.57805	57.60687	0.009801	0.013201	
197	0.3605904	0.6394096	57.57805	57.60687	0.009801	0.013201	
426	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
180	0.360591	0.639409	57.57805	57.60687	0.009801	0.013201	
227	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
231	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
232	0.3605907	0.6394093	57.57805	57.60687	0.009801	0.013201	
221	0.3605909	0.6394091	57.57805	57.60687	0.009801	0.013201	
253	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
250	0.3605902	0.6394098	57.57805	57.60687	0.009801	0.013201	
189	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
209	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
429	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
217	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
361	0.3606452	0.6393548	57.57805	57.60687	0.009801	0.0132	
212	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
386	0.3606453	0.6393547	57.57805	57.60687	0.009801	0.0132	
213	0.3605906	0.6394094	57.57805	57.60687	0.009801	0.013201	
216	0.3605906	0.6394094	57.57805	57.60687	0.009801	0.013201	
176	0.360591	0.639409	57.57805	57.60687	0.009801	0.013201	
242	0.3606446	0.6393554	57.57805	57.60687	0.009801	0.0132	
419	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
247	0.3605901	0.6394099	57.57805	57.60687	0.009801	0.013201	
203	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
229	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
199	0.3605902	0.6394098	57.57805	57.60687	0.009801	0.013201	
222	0.3605903	0.6394097	57.57805	57.60687	0.009801	0.013201	
316	0.3606449	0.6393551	57.57805	57.60687	0.009801	0.0132	
165	0.3605911	0.6394089	57.57805	57.60687	0.009801	0.013201	
259	0.3606448	0.6393552	57.57805	57.60687	0.009801	0.0132	
205	0.3606454	0.6393546	57.57805	57.60687	0.009801	0.0132	
256	0.3606456	0.6393544	57.57805	57.60687	0.009801	0.0132	
235	0.3606453	0.6393547	57.57805	57.60687	0.009801	0.0132	
187	0.3605905	0.6394095	57.57805	57.60687	0.009801	0.013201	
216	0.3606455	0.6393545	57.57805	57.60687	0.009801	0.0132	

Bibliography

[1] T. Benaglia et. al, "mixtools: An R Package for Analyzing Finite Mixture Models," *Journal of Statistical Software*, vol. 32, Issue 6 pp 1-29, Oct. 2009

[2] J.A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," International Computer Science Institute, Berkley, CA. Rep: TR-97-021, 1998.

[3] C.M. Bishop, "Mixture Models and EM," in Pattern Recognition and Machine Learning, New York: Springer, 2006, ch. 9, sec. 2, pp.423-459.

[4] L. Cheng et. al, "The Impact of Contract Manufacturing on Inventory Performance: An Examination of U.S. Manufacturing Industries," *Decision Sciences*, vol. 43, no 5, pp.889-928, Oct. 2012.

[5] D.C. Montgomery, "Control Charts for Variables," in *Introduction to Statistical Quality Control, 2nd ed*. New York: J.W. & Sons, 1991, ch. 6, sec. 2, pp.203-230.

[6] R.K. Pearson, "Dealing with Missing Data, in *Exploring Data in Engineering, the Sciences, and Medicine*. New York: Oxford University Press, 2011. ch. 16, sec. 16.7, pp 723 - 735

[7] R Documentation, "Performs Parametric Bootstrap for Sequentially Testing the Number of Components in Various Mixture Models", in boot.comp {mixtools}, [Online]. Available: http://127.0.0.1:22069/library/mixtools/html/boot.comp.html

Vita

Wesley Freeman is a Quality Specialist for a Fortune 500 company in Austin, Texas. His work for the Quality Organization consist of providing analysis of manufacturing process data, designs of experiments, and other statistical analysis. These activities are to support manufacturing organizations, root cause analysis for customer complaints, and product design creation.

This report was typed by James Wesley Freeman